PAPER NAME

**Face recognition using Tensorflow.docx**

AUTHOR

**DIVVELA VISHNU SAI KUMAR**

WORD COUNT

**4363 Words**

CHARACTER COUNT

**22272 Characters**

PAGE COUNT

**19 Pages**

FILE SIZE

**1.2MB**

SUBMISSION DATE

**Nov 8, 2022 7:17 AM GMT+5:30**

REPORT DATE

**Nov 8, 2022 7:18 AM GMT+5:30**

## 50% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 41% Internet database
- 8% Publications database
- Crossref database
- Crossref Posted Content database
- 44% Submitted Works database

## Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 14 words)

# Face Recognition using Tensorflow

## ABSTRACT

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from noisy, structured, and unstructured data and to apply knowledge from data in a wide variety of application domains. Data science is related to data mining, machine learning and big data.

Data science is "the concept of unifying statistics, data analysis, computer science, and their related methods" to "understand and analyze real-world phenomena" with data. It uses techniques and theories drawn from many fields in the context of mathematics, statistics, computer science, information science and domain knowledge. However, data science is different from computer science and information science. Turing Award winner Jim Gray envisioned data science as the "fourth paradigm" of science (empirical, theoretical, computational, and now data-driven) and argued that "everything about science is changing because of the impact of information technology" and the deluge of data. .

The field of the Machine Learning can also be called as enabling or controlling the computers which makes their predictions successfully by using their past experiences, it has a successful development with the help of increase in the capacity of storage rapidly and processing of the computers power. The methods of Machine Learning had been employed widely in bioinformation also. There are difficulties and much cost in analysing biologically, and it has the development of Machine Learning sophisticatedly approaching for this area.

In this report we are going to know about the fundamental topics for Machine Learning , like Feature engineering, types of classification of Machine learning , applications related to the Machine learning, history related to Machine Learning, and we will learn about deep learning which is a subset of Machine learning, Tools of deep learning, what is perceptron, structure of neural network, Activation functions, Optimizers, CNN, RNN and finally we will know about Face recognition using TensorFlow.

Deep learning is a subset of machine learning that uses mathematical functions to map input to output. These functions can extract non-redundant information or patterns from data, allowing them to establish a relationship between input and output. This is known as learning and the process of learning is called training.

## Introduction to Machine learning

The Machine Learning term was created by a scientist 'Arthur Samuel' in the year of 1959. He is an American who works in the field of Computer gaming and AI. And he had stated that "AI gives the computers an ability for learning without programming explicitly".

In the year of 1997, an scientist called 'Tom Mitchell' given a mathematical relational type definition that is "an computers program learns from the experience 'E' with something task called 'T' which is measured by 'P' and it again improves the experience 'E'. "

ML can be called as the one of the best interesting subfields in Artificial intelligence and Computer science also.

If we take an example called throwing a ball for understanding about ML briefly, for example in first attempt we will come to known that we must apply an much force on it, after completion of second attempt we will come to know that we have change some throw angle in it to reach the target. Here what is happening is after completion of every attempt we are learning something, and we are trying to improve our self. That means we are programmed to learn from our experience for better result.

This example follows the proposal of Alan Turing that "Computing Machines and their Intelligence" in which a question that "can machines think "can be replaced with the question "can Machines do what we can do"

When we consider the field of Data analytics, ML is used to solve the tuff or complex problems and Algorithms that are lead to Predictions by themselves. This is known as predictive analytics. These analytic models can be useful for researchers to produce 'reliable, repeatable decisions, and results.

For example, you are decided to go for a vacation, we will go through any travel agencies websites, and we will search an hotel for vacation, when you check for hotels at down you will see a specified title that, "you may also like these hotels!". This is an example of Machine Learning.

Suppose if you want to create a program of predicting the traffic signals patterns, in the case of busy Intersection called task's' the Machine Learning data with algorithm learns from the past traffic patterns called experience 'E'. If the program learned perfectly from past, it will perform perfectly in future predictions also called performances''.

There are so many highly complex real world problems in the world, by inventing specialized algorithms, they will solve them perfectly every time. The example complex problems which can be included in Machine Learning are "is this cancer?", "will that person likes sports?". The problems like these models are excellent targets for ML.

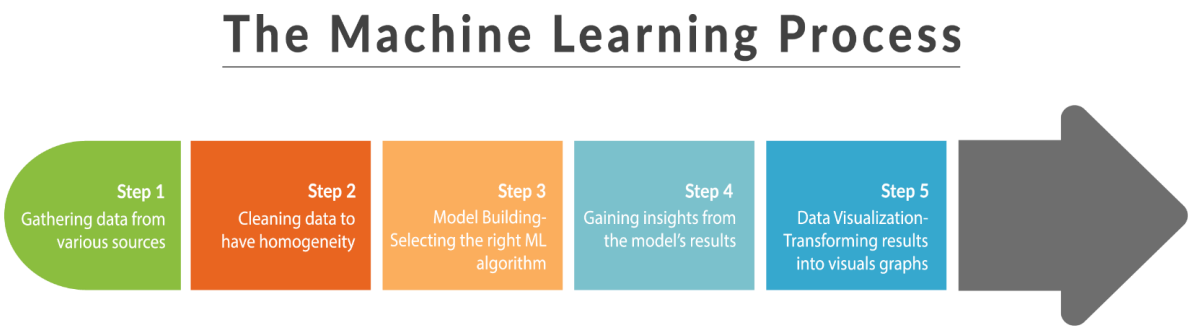


**Fig1.** Steps in Machine Learning

## Classification Of Machine Learning.

There are three main types of classifications in MACHINE LEARNING. These classifications are classified based on the nature of the learning, which are as following:


**Fig2.** Classification of Machine learning

## 1)SUPERVISED Learning:

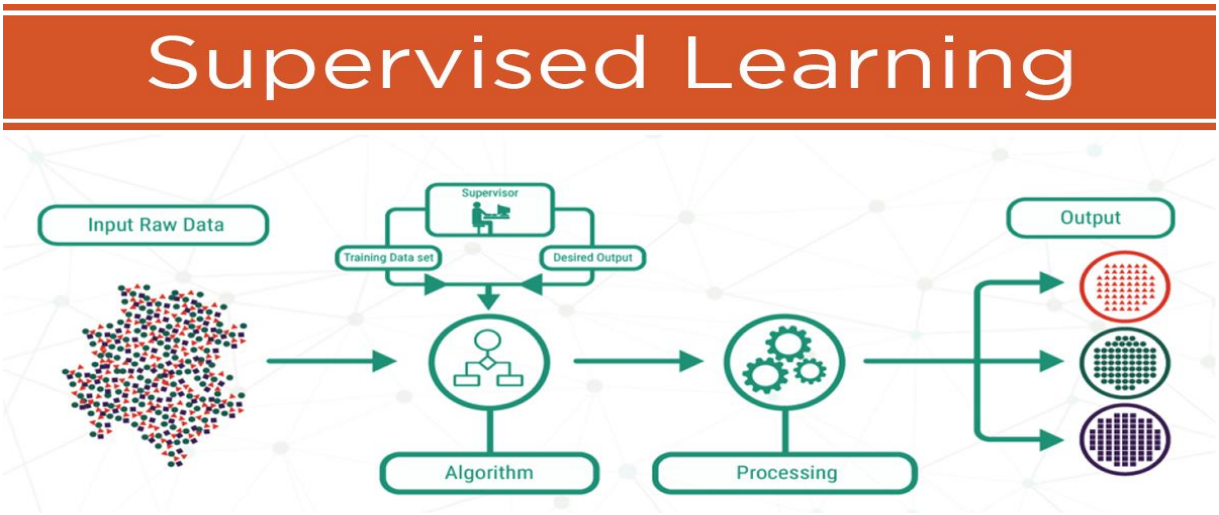


**Fig3.** Supervised learning process

Supervised Learning is also known as for example take a function called Y=f(X), where x is called input variables and y is called an output variable. The above function is an algorithm. It learns to map the function from input to the output.

The goal of this algorithm is to map the above function so that if we give an new input data (x) we can get the output variables (y) for that given data.

This type of process is known as Supervised Learning because the algorithms process learns from the data can called as teacher, the algorithms automatically make predictions on the training data, and it will be corrected by teacher. This Learning will stop when it reaches the perfect level of performance.

It is also known as labelled data learning. If we give the input to the computer as the weight of one-rupee Indian currency is 3 grams, French currency is 7grams and in German currency is 5 grams. Now if we give an empty coin of 3 grams it identifies as one-rupee Indian currency.

## 2)UNSUPERVISED Learning:

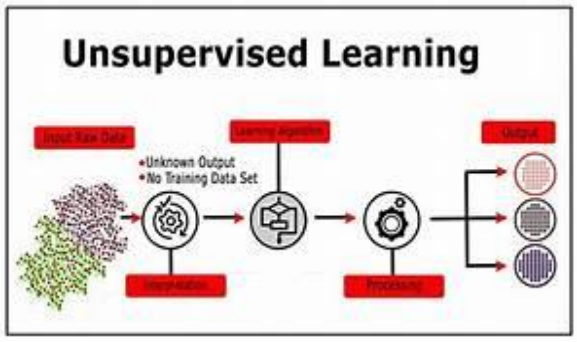


**Fig4.** Unsupervised machine learning process

Unsupervised learning also known as when we consider a algorithm, if it trying to learn plain examples without any type of response, and again leaving to the algorithm to find out or to determine the Data patterns on its own. These types of algorithms can be tends to redesign or restructure the data into something else like it may represent the new series of Un correlated value.

It is also known as unlabeled data learning because it clusters the given input data. It will give an unknown output and it don't contain training data set. It will resemble with the methods of human beings to figure out certain objects, for example observation of degree of similarity between the two objects. Marketing automation which is an example of Recommendation system is based on the Unsupervised Learning.

## 3)REINFORCEMENT Learning:

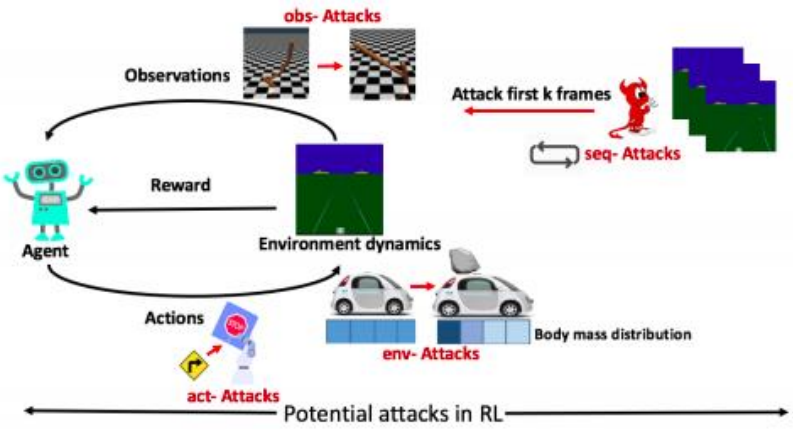




**Fig5.** Steps in Reinforcement learning

At the point when you present the calculation with models that need names, as in solo learning. Be that as it may, you can go with a model with positive or negative input by the arrangement the calculation proposes goes under the classification of Reinforcement realizing, which is associated with applications for which the calculation must decide (so the

item is prescriptive, not only expressive, as in unaided learning), and the choices bear results. In the human world, it is much the same as learning by experimentation.

In this type of Learning it learns from the errors. Because it is added with an penalty. So we can say that errors help you to learn. These penalties can add Cost, Loss of Time,Pain e.t.c., this Learning works on the princple of feedback, a best example for Reinforcement Learning is computer playing the video games by itself.

Simply reinforcement learning is a reward base learning or it works on the principle of feedback. For example if we give dog image to the computer if it gives the output as it is cat then we have give feedback that no it is a dog, after that give another type dog image. It learns and give that it is a dog. This is called reinforcement learning.

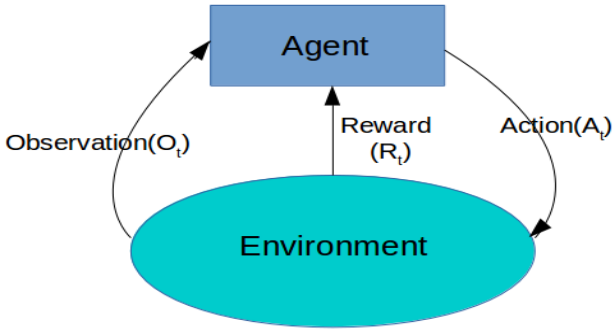


**Fig6.** Working of a agent in reinforcement learning

## Introduction to Deep Learning

Deep learning is a branch of machine learning that is completely based on artificial neural networks, because the neural network will imitate the human brain, so deep learning is also a kind of imitation of the human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for several years. It's all the rage these days because before we didn't have so much computing power and a lot of data. As computing power increases exponentially over the past 20 years, deep learning and machine learning have entered the scene.

Deep learning is a special kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts and more abstract representations computed in terms of less abstract ones.

In the human brain, with approximately 100 billion neurons combined, it is a picture of an individual neuron, and each neuron is connected to over a thousand of its neighbours.

The question is how we recreate these neurons in a computer. So we create an artificial structure called artificial neural network where we have nodes or neurons. We have some neurons for the input value and some for the output value, and in between there may be lots of neurons connected in the hidden layer.

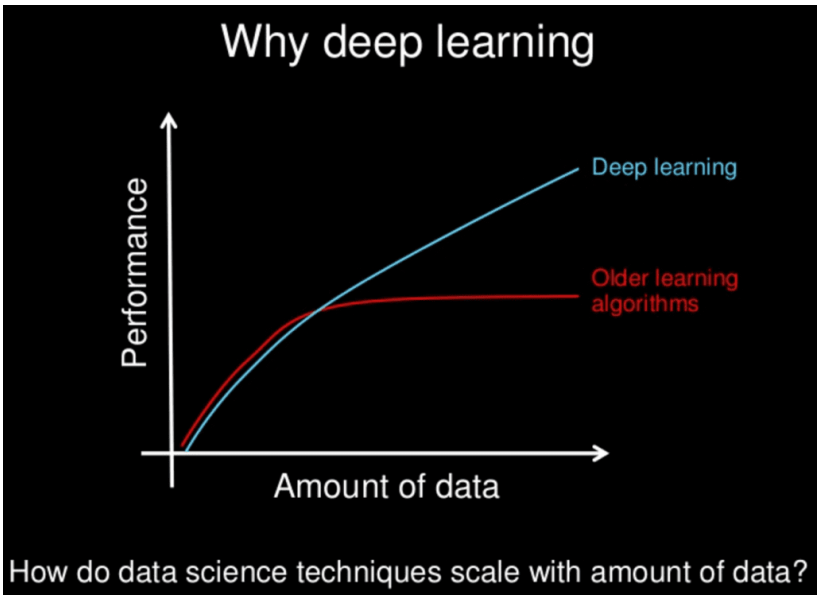


**Fig7.** Comparison of deep learning and other algorithms

## Perceptron

The Perceptron was introduced by Frank Rosenblatt in 1957. He proposed a Perceptron learning rule based on the original MCP neuron. Perceptron is an algorithm for supervised learning of binary classifiers. This algorithm allows neurons to learn and process the elements in the training set one at a time.

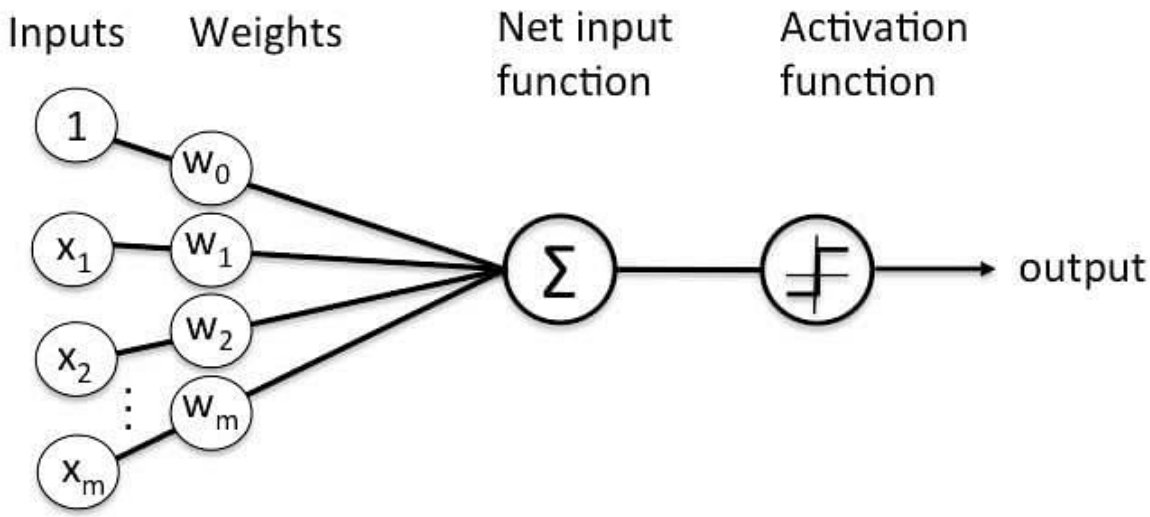


**Fig8.** Working of a perceptron

Perceptron is classified into two types; those are Single layer perceptron and Multi-layer perceptron

Single layer - Single layer perceptron's can learn only linearly separable patterns

Multilayer - Multilayer perceptron's or feedforward neural networks with two or more layers have the greater processing power which is also known as a complete neural network.

The Perceptron algorithm learns the weights for the input signals in order to draw a linear decision boundary. This enables you to distinguish between the two linearly separable classes +1 and -1.

Perceptron Learning Rule states that the algorithm would automatically learn the optimal weight coefficients. The input features are then multiplied with these weights to determine if a neuron fires or not.

## Layers in Neural Network

A neural network is built by layers which will be appended like at stack. In the below given image the vertically dotted lines represent the layers. There are total three types of layers in a neural network.

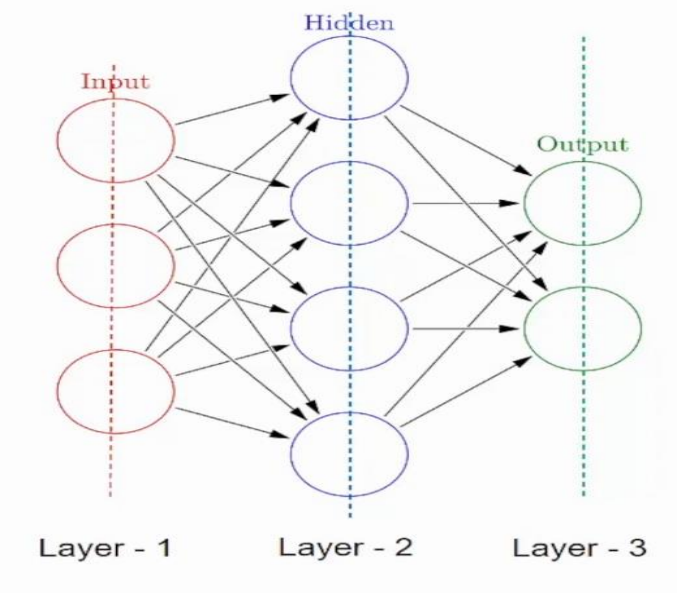


**Fig9.** Layers in a neural network

Input Layer– First is the input layer. This layer will accept the data and pass it to the rest of the network.

Hidden Layer – The second type of layer is called the hidden layer. A neural network has either one or more hidden layers. In the above case, the number is 1. The hidden layers are responsible for the superior performance and complexity of neural networks. They perform multiple functions at the same time, such as data transformation, automatic element creation, etc.

Output Layer – The last type of layer is the output layer. The output layer contains the result or output of the problem. The raw images are passed to the input layer and we receive the output in the output layer.

## Activation Functions in Neural networks

Now that we know how a neural network combines different inputs using weights, let's move on to the last aspect of a neuron called the activation function. So far we've simply added some weighted inputs and calculated some output, and that output can read from minus infinity to infinity.

However, this can be challenged in many circumstances. Suppose we first want to estimate a person's age from their height, weight, and cholesterol level, and then classify the person as old or not based on whether they are older than 60 years.

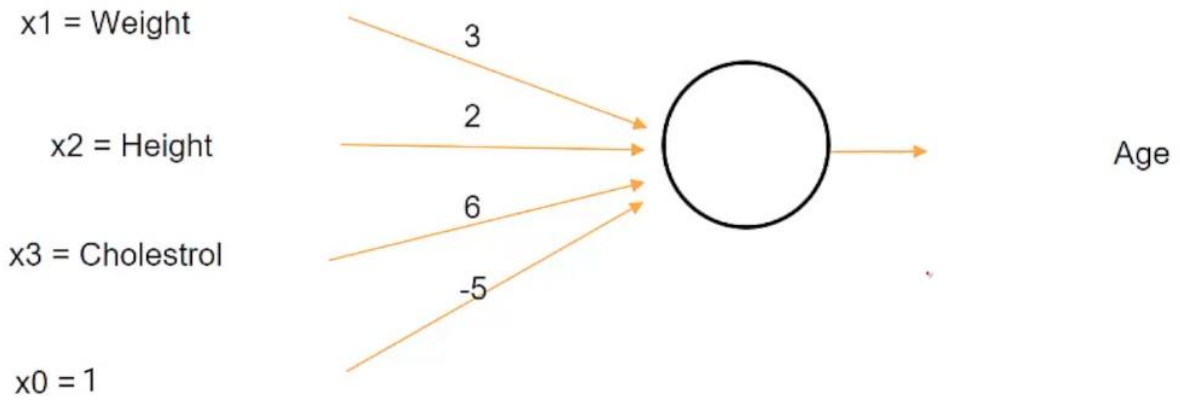


**Fig10.** Working of a node in neural network

Now, if we use this given neuron, an age of -20 is even possible. You know that the age range according to the current structure of this neuron will be from -∞ to ∞. So even someone's age of -20 is possible, given this absurd range for age we can still use our status to decide if a person is old or not. For example, if we said a certain criterion like a person is old, only if the age is more than 60 years. So even if the age comes out to -20, we can use this criterion to classify the person as ageless.

Depending on the type of transformation needed, there can be different kinds of activation functions. Let's take a look at some popular activation features –

**1.Sigmoid Activation Function**

This function transforms the range of combined inputs to a range between 0 and 1. For example, if the output is from minus infinity to infinity, which is represented by the x-axis, the sigmoid function will limit that infinity. range to a value between 0 and 1.

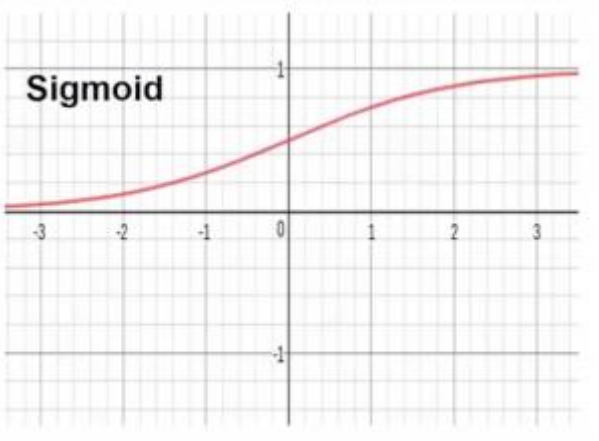


**Fig11.** Sigmoid activation function

**2. Tanh Activation Function**- This function transforms the range of combined inputs to a range between -1 and 1. Tanh looks very similar to a sigmoid shape, but limits the range to between -1 and 1.

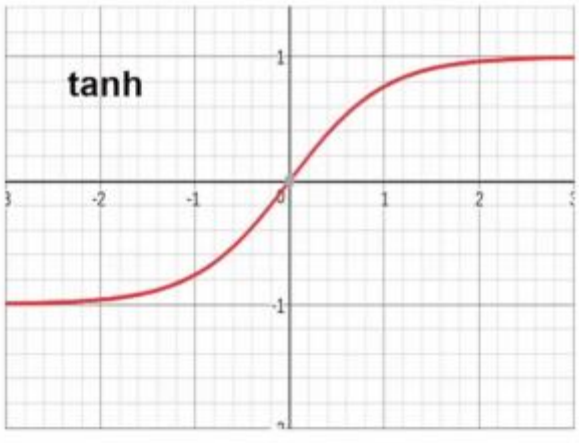


**Fig12.** Tanh activation function graph

### 3. Activation function ReLU (Rectified Linear Unit).

ReLU is currently the most used activation function in the world. Since then, it has been used in almost all convolutional neural networks or deep learning.

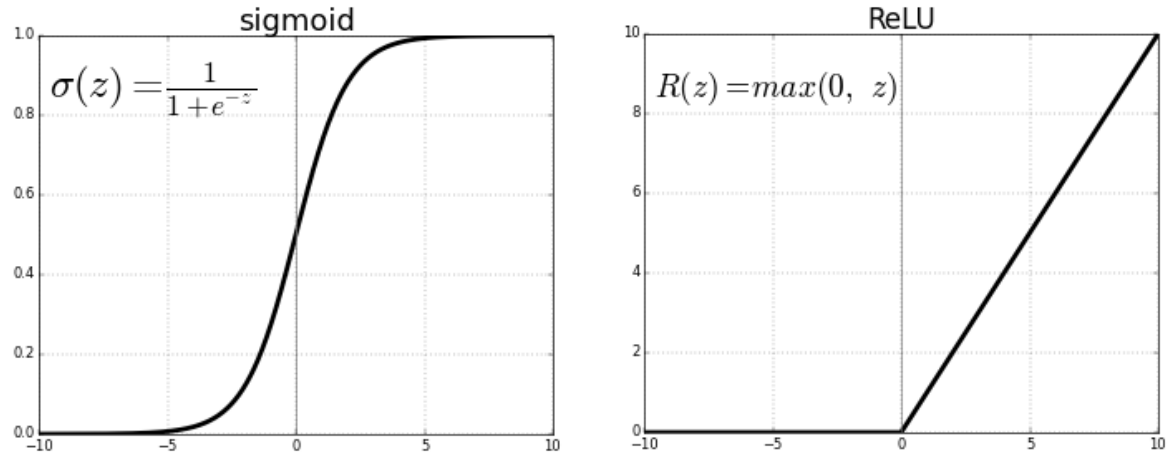


**Fig13.** Comparison of sigmoid and relu graphs

### 4. Leaky ReLU

It is an attempt to solve the dying ReLU problem. In Relu in the case of negative value it will just give the output as zero, so there will be no updation in the weights. So Because of Relu activation function there is no use of Back propagation. To solve this Leaky Relu is developed in which this function will accept a less range of negative value.

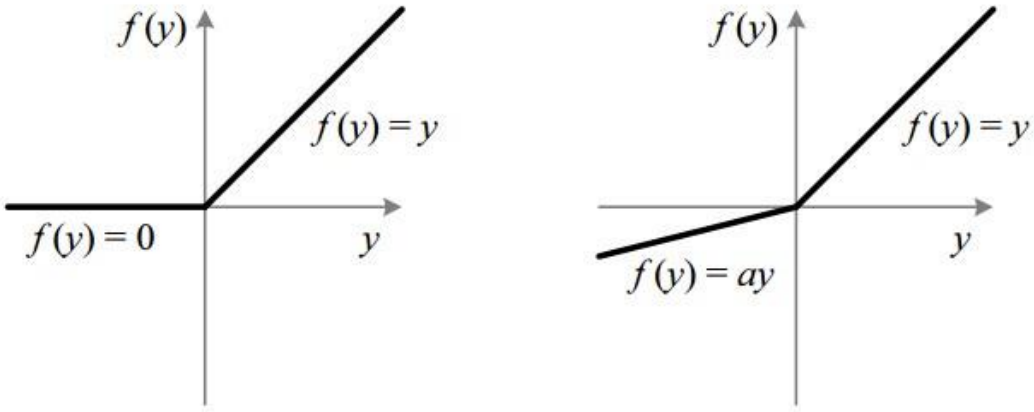


**Fig14.** Leaky relu activation function

### 5. Threshold Activation Function

This activation function will works as if x >0 it will produce 1 as output otherwise it will produce 0 as output.

### 6. Softmax Activation Function

Softmax is an activation function that scales numbers/logits into probabilities. The output of a Softmax is a vector (say v) with probabilities of each possible outcome. The probabilities in vector v sums to one for all possible outcomes or classes. Mathematically, Softmax is defined as,

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)}$$

where,

| | |
|---|---|
| $y$ | is an input vector to a softmax function, S. It consist of $n$ elements for $n$ classes (possible outcomes) |
| $y_i$ | the $i$-th element of the input vector. It can take any value between -inf and +inf |
| $exp(y_i)$ | standard exponential function applied on $y_i$. The result is a small value (close to 0 but never 0) if $y_i < 0$ and a large value if $y_i$ is large. eg<br>• $\exp(55) = 7.69e{+}23$ (A very large value)<br>• $\exp(-55) = 1.30e\text{-}24$ (A very small value close to 0)<br>**Note**: $\exp(*)$ is just $\mathbf{e}^*$ where $\mathbf{e} = 2.718$, the Euler's number. |
| $\sum_{j=1}^{n} \exp(y_j)$ | A normalization term. It ensures that the values of output vector $S(y)_i$ sums to 1 for $i$-th class and each of them and each of them is in the range 0 and 1 which makes up a valid probability distribution. |
| n | Number of classes (possible outcomes) |

**Fig15.** Formula of softmax function

## Optimizers for Neural Network

Optimizers are defined as they are the approach used in the process of Backpropagation and the Backpropagation means updating the weights and bias values of the neural networks in training process with respect to loss of neural network.

How you should change your weights or the learning rate of your neural network to reduce losses is defined by the optimizers you use. Optimization algorithms or strategies are responsible for reducing losses and providing the most accurate results possible.

Types of optimizers

### 1) Gradient descent

Gradient Descent is the most basic but most used optimization algorithm. It is widely used in linear regression and classification algorithms. Backpropagation in neural networks also uses the gradient descent algorithm.

Gradient descent is a first-order optimization algorithm that depends on the derivative of the first-order loss function. It calculates how the weights should be changed so that the function can reach a minimum. Through backpropagation, the loss is transferred from one layer to another and the model parameters, also known as weights, are adjusted depending on the losses so that the loss can be minimized.

algorithm: $\theta = \theta - \alpha \cdot \nabla J(\theta)$

### 2) Stochastic gradient descent

It is a variant of Gradient Descent. It tries to update the model parameters more often. In this, the model parameters are changed after calculating the loss on each training example. So if the dataset contains 1000 rows, SGD updates the model parameters 1000 times in one dataset cycle instead of once as in Gradient Descent.

### 3) Mini-Batch Gradient Descent

It is the best of all variants of gradient descent algorithms. It is an improvement on both SGD and standard descent. Updates model parameters after each batch. Thus, the dataset is divided into different batches and after each batch the parameters are updated.

### 4) Momentum

Momentum was invented to reduce high variance in SGD and soften convergence. It accelerates the convergence to the relevant direction and reduces the fluctuation to the irrelevant direction. In this method, one more hyperparameter known as momentum symbolized by "γ" is used.

$V(t) = \gamma V(t-1) + \alpha . \nabla J(\theta)$

## 5) Adagrad

One of the disadvantages of all explained optimizers is that the learning rate is constant for all parameters and for each cycle. This optimizer changes the learning speed. It varies the learning rate 'η' for each parameter and at each time step 't'. It is a type of second-order optimization algorithm. It works on the derivative of the error function.

The derivative of the loss function for the given parameters at the given time t

## 6) AdaDelta

It is an AdaGrad extension that tends to eliminate the learning decay problem. Instead of accumulating all previously scaled transitions, Adadelta limits the window of accumulated past transitions to some fixed size w. In this, an exponential moving average is used rather than the sum of all gradients.

$E[g^2](t)=\gamma.E[g^2](t-1)+(1-\gamma).g^2(t)$

## 7) Adam

Adam (Adaptive Moment Estimation) works with first- and second-order momenta. The intuition behind Adam is that we don't want to scroll that fast just so we can skip the minimum, but we want to slow down a bit for careful searching. In addition to storing the exponentially decreasing mean of past quadratic gradients like AdaDelta, Adam also stores the exponentially decreasing mean of past gradients M(t).

M(t) and V(t) are the values of the first moment, which is the mean value, and the second moment, which is the uncentered variance of the gradients.

## Tools Used for Implementing Neural Networks

### 1. TensorFlow

TensorFlow is a deep learning tool that was written in highly optimized C++ and CUDA (Nvidia's GPU programming language) and provides interfaces to languages like Python, Java, Go. It is an open source library developed by the tech giant Google to run deep learning applications smoothly.

TensorFlow makes it easy for beginners and even experts to build machine learning models for mobile, web, desktop and cloud.

It is also used to create large scale neural networks with multiple layers. If you want to solve deep learning or machine learning problems such as classification, perception, understanding, discovery, prediction, and creation, TensorFlow is the right deep learning tool for you.

### 2. Keras

Keras is a high-level neural network API that is capable of running on top of TensorFlow or Theano. It is written in Python and was developed mainly to enable faster experimentation. The Keras deep learning library makes prototyping easier and faster for the user using modularity, minimalism and easy extensibility.

Keras is a deep learning tool that supports recurrent networks and convolutional networks individually and in combination of both. It also supports training with multiple inputs and outputs. It follows best practices to reduce cognitive load by offering consistent and simple APIs. In addition, it minimizes the number of user actions needed for common use cases and provides clear feedback when any error is detected.

### 3. Torch

Torch is a powerful open source program that uses the LuaJIT scripting language and C/CUDA implementation. If you use this deep learning tool, you will be able to take advantage of its powerful features such as: multiple routines for indexing, transposing, slicing, amazing interface to C via LuaJIT, neural network. It offers fast and efficient GPU support and is easy to embed, making it easy to work with iOS, Android, FPGA.

### 4. H2O.ai

H2O is a deep learning tool that was built from the ground up in Java and builds on seamless integration with other open source products such as Apache Hadoop and Spark. It has an easy-to-use web-based user interface and is massively scalable in big data analytics.

H2O, an open source deep learning tool, supports the most widely used machine learning algorithms. It is a fast, scalable machine learning application interface used for deep learning, elastic network, logistic regression, gradient boosting; to name a few. H20 makes it easier for anyone to use machine learning algorithms and predictive analytics to solve any business problem.

### 5. DeepLearningKit

DeepLearningKit is an open source deep learning tool for Apple iOS, OS X, tvOS, etc. The main idea behind its creation was to support pre-trained models on all Apple devices that have a GPU. This deep learning tool is developed in Swift and can be used on GPU devices to perform deep learning calculations with low latency. DeepLearningKit also supports convolutional neural networks. His vision is to support other deep learning tools like Torch and TensorFlow.

## Face recognition using Tensorflow

Facial recognition is a method of identifying or confirming a person's identity based on their face. Facial recognition systems can be used to identify people in photos, videos, or in real time.

Facial recognition is a category of biometric security. Other forms of biometric software include voice recognition, fingerprint recognition, and retinal or iris recognition. The technology is primarily used for security and law enforcement purposes, but there is growing interest in other application areas.

Many people are familiar with the FaceID facial recognition technology used to unlock iPhones (although this is just a facial recognition application). Rather than relying on a huge database of photos to determine an individual's identity, facial recognition typically simply identifies and recognizes the individual as the sole owner of the device, limiting access to other users.

In addition to unlocking phones, facial recognition works by matching the faces of people who pass in front of a special camera to images of people on your watchlist. Watchlists can include images of anyone, even those who are not suspected of cheating, and the images can come from anywhere, even from their social media accounts. Facial technology systems may vary, but generally work as follows:

Step 1: Face Detection The

camera recognizes and locates images of faces, either alone or in a crowd. The image can show a person facing front or profile.

Step 2: Face Analysis

Next, facial images are captured and analyzed. Most facial recognition technologies are based on 2D images rather than 3D images. This is because 2D images can be compared more conveniently with public photos or photos in a database. The software reads the shape of your face. Important factors include the distance between the eyes, the depth of the eye sockets, the distance from the forehead to the chin, the shape of the cheekbones, the contours of the lips, ears and chin.

The goal is to identify key facial features to identify faces.

Step 3: Convert Image to Data The

face capture process converts analog information (the face) into a series of digital information (data) based on the person's facial features. Facial analysis essentially turns into a mathematical formula. Numeric codes are called faceprints. Each person has a unique facial print, just like their thumbprint is unique.

Step 4: Find Matches

Next, compare the headshot to a database of other known faces. For example, the FBI has access to up to 650 million photos from various government databases. On Facebook, any photo tagged with a person's name becomes part of Facebook's database and can also be used for facial recognition. If the face print matches the image in the facial recognition database, a decision is made of all biometric measurements, facial recognition is considered the most natural. Intuitively, this makes sense, since we usually recognize ourselves and others by our faces rather than our thumbprints or irises. It is estimated that more than half of the world's population are exposed to facial recognition technology on a regular basis.

In this project we are using a architecture called Siamese network, this network will take two images for training process with labels 0 or 1. We will make three datasets one will contain our image with different angles, second one will contain our images only same as first one, third one contain the images of other people. All these datasets have 500 images each. Now randomly we will take the image from first dataset and combine with second and third datasets, if the first image and second image are belonging to same person then we will give the label as 1 otherwise we will give the label as 0. Like this we will make a dataset and we use keras sub modelling and custom training loop topics to create our model which works same as the Siamese model and we will start the training process. After training completes, we can use the same model for face recognition, first we have to collect the images of the authorized user and we will store it in a folder, after that we will take a image input and send it with every image the authorized person has saved and if it is having atleast 80% threshold then it will print as person is authorized otherwise it will print it as not authorized.

## Results

```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    frame = frame[30:30+250,220:220+250,:]

    cv2.imshow('Verification', frame)

    # Verification trigger
    if cv2.waitKey(10) & 0xFF == ord('v'):
        # Save input image to application_data/input_image folder
#         hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
#         h, s, v = cv2.split(hsv)

#         lim = 255 - 10
#         v[v > lim] = 255
#         v[v <= lim] -= 10

#         final_hsv = cv2.merge((h, s, v))
#         img = cv2.cvtColor(final_hsv, cv2.COLOR_HSV2BGR)

        cv2.imwrite(os.path.join('application_data', 'input_image', 'input_image.jpg'), frame)
        # Run verification
        results, verified = verify(model, 0.8, 0.7)
        print(verified)

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

```
False
```

**Fig16**. Output of the face recognition model

## Conclusion:

Facial recognition is a method of identifying or confirming a person's identity based on their face. Facial recognition systems can be used to identify people in photos, videos, or in real time.

Facial recognition is a category of biometric security. Other forms of biometric software include voice recognition, fingerprint recognition, and retinal or iris recognition. The technology is primarily used for security and law enforcement purposes, but there is growing interest in other application areas.

## 50% Overall Similarity

Top sources found in the following databases:

- 41% Internet database
- Crossref database
- 44% Submitted Works database
- 8% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | |
|---|---|---|
| 1 | analyticsvidhya.com<br>Internet | 7% |
| 2 | Midlands State University on 2022-08-11<br>Submitted works | 7% |
| 3 | CVC Nigeria Consortium on 2022-04-08<br>Submitted works | 5% |
| 4 | geeksforgeeks.org<br>Internet | 3% |
| 5 | Washington University in St. Louis on 2022-02-04<br>Submitted works | 2% |
| 6 | University of Huddersfield on 2022-04-01<br>Submitted works | 2% |
| 7 | University of York on 2022-08-21<br>Submitted works | 2% |
| 8 | towardsdatascience.com<br>Internet | 2% |

| | | |
|---|---|---|
| 9 | SVKM International School on 2020-08-12<br>Submitted works | 2% |
| 10 | University of Sunderland on 2022-04-08<br>Submitted works | 2% |
| 11 | Meerut Institute of Engineering & Technology on 2020-05-15<br>Submitted works | 1% |
| 12 | University of Wales Institute, Cardiff on 2020-01-04<br>Submitted works | 1% |
| 13 | i2tutorials.com<br>Internet | 1% |
| 14 | ACT Information Consulting on 2022-01-28<br>Submitted works | 1% |
| 15 | Universiti Teknologi Malaysia on 2021-09-17<br>Submitted works | 1% |
| 16 | v7labs.com<br>Internet | 1% |
| 17 | Brigham Young University on 2021-05-18<br>Submitted works | 1% |
| 18 | askingthelot.com<br>Internet | 1% |
| 19 | medium.com<br>Internet | 1% |
| 20 | University of Computer Studies on 2020-06-29<br>Submitted works | <1% |

| | | |
|---|---|---|
| 21 | coursehero.com<br>Internet | <1% |
| 22 | UC, Boulder on 2019-12-13<br>Submitted works | <1% |
| 23 | jozilla.net<br>Internet | <1% |
| 24 | Magdalen College School on 2017-06-15<br>Submitted works | <1% |
| 25 | Colorado Technical University on 2021-03-03<br>Submitted works | <1% |
| 26 | Nexford University on 2021-05-31<br>Submitted works | <1% |
| 27 | Central Queensland University on 2018-06-07<br>Submitted works | <1% |
| 28 | Delhi Technological University on 2018-07-17<br>Submitted works | <1% |
| 29 | University of Bedfordshire on 2022-11-04<br>Submitted works | <1% |
| 30 | ijasret.com<br>Internet | <1% |
| 31 | saarj.com<br>Internet | <1% |